Security of Systems on Chip

Eda Tumer[1], Dr. Leandros Maglaras[1]
[1]De Montfort University, School of Computer Science and Informatics, Leicester, UK
ed.tumer@gmail.com
leandros.maglaras@dmu.ac.uk

Abstract: In recent years, technology has started to evolve to become more power efficient, powerful in terms of processors and smaller in size. This evolution of electronics has led microprocessors and other components to be merged to form a circuit called System-on-Chip. If we are to make a vast and cursory comparison between SoC and microcontrollers, microprocessors and CPUs; we would come to the conclusion of SoCs being a single chip, doing all the things the other components can do yet without needing any external parts. So SoCs are computers just by themselves. Furthermore, SoCs have more memory than microcontroller in general. Being a computer just by themselves allows them also to become servers. Nowadays, an SoC may be regarded also as a Server-on-Chip (Davis, 2012).

Keywords: Cyber Security; System on Chip; Microcontroller Security; Hardware Security.

1. Introduction

Briefly, an SoC is an integrated circuit(IC) viz. a chip (Rajesvari et al, 2013) which is, in other words, a wide-ranging integration of various components such as processor cores, accelerators, peripheral controllers, memory controllers, intellectual properties(IP) along with on-chip and off-chip memory components on only one IC for specialised uses (Davis, 2012)(Wang et al, 2008). As long as Moore's Law (Gelsinger, 2006) continues on, the chips and its parts will shrink further enabling more transistors to be put on the same part of the silicon (Davis, 2012).

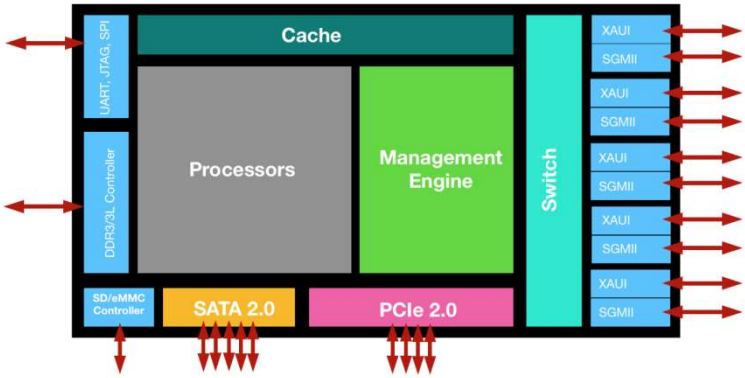

Figure 1. System on Chip Properties

The Internet of Things is the interconnection between intelligent devices and systems, and data collected by physical sensors, actuators and other physical objects (Nolan, n.d.)(GSM Association, 2014). These devices process and report data to the network and some of them provide outputs which control other devices or systems, either autonomously or by network commands (Nolan, n.d.).

In simpler words, the IoT is a system of devices that are interconnected. Furthermore they can control each other to mitigate changes in the environment which are provided by sensors. The controllers of these devices can connect to the internet in order to provide distant control to users.

2. Role of SoC and Microcontrollers in IoT Systems

After the invention of the first silicon chip in 1958 by Jack Kilby, the chips diminished in size. When a whole system was integrated on a single chip in 1970, it led to huge downsizing of physical systems. With these ad-

vancements, implementing a system started to require less power and space. It also grants higher performance due to increased number of circuits on the chip, and further reliability (Palomar Technologies, n.d.). Subsequent to these advantages, it was inevitable that many systems started to use SoC, though some continue to use microcontrollers still.

As the needs and demands for systems get complicated, so do the systems' operations and structures. The operations and functions of these systems naturally depend on the signals of the components and calculations or decisions made by the brains of this system, namely microcontrollers and SoCs. In a case that these components are cornerstones of the system, they may create single-point of failures if not protected and handled with redundancy. Since these MCUs and SoCs are interconnected, they also expose vulnerability risks in the whole system. Therefore they have to be protected against possible cyber attacks.

Due to the reason that the concept of IoT is growing exponentially (as seen on Figure 2), the need for protection of these systems become more and more crucial with the same rate (Maglaras et al, 2018). The full range of potential applications of IoT devices has not yet been perceived and its application range includes infrastructure, personal, medical, industrial and more (Nolan, n.d.). In order to provide the highest level of integration and save space, many IoT devices are thought to have a single SoC (Nolan, n.d.).

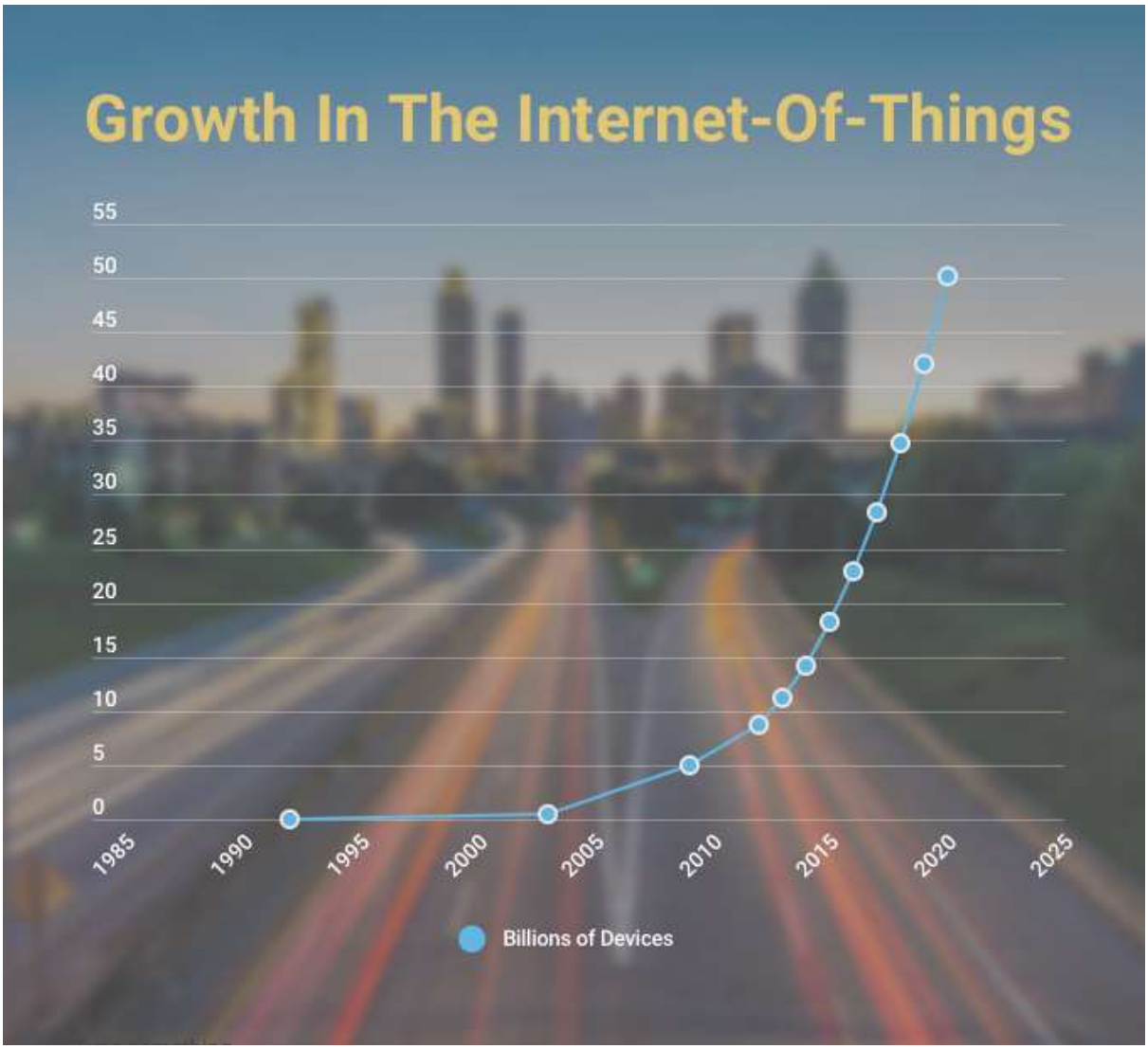

Figure 2. Growth of IoT (NCTA, 2015)

Because IoT systems are a network of many connected devices, the process of securing the whole system also becomes complex and challenging. There can be malicious hardware or software which may compromise the security of this ecosystem. Security issues of IoT systems are to be handled with even more care because the data transmitted in the network between the nodes may be sensitive, ranging from personal and private data to secrets of an organisation (Kumar et al, 2017).

`

To implement the system in a secure way, the organisations have to consider potential adversaries. It is described to be a very creative process by (Ray and Jin, 2015) because the team has to think of every possible scenario and considerations of immoral actions and leakages. The following adversary examples are self-explanatory hence only their titles are mentioned.

— Unprivileged Software Adversary, System Software Adversary, Software Covert Channel Adversary (when a side-channel or covert-channel has access to non-functional characteristics of the system) (Ray and Jin, 2015)

— Naive Hardware Adversary (when attackers manage to get into hardware devices), Hardware Reverse-Engineering Adversary, Malicious Hardware Intrusion Adversary (Ray and Jin, 2015)

3. Hardware Security of SoC

As a consequence of the rapid growth of IoT (NCTA, 2015)(Evans, 2011), many edge devices have to be given access to a large number of security assets which are to be protected from unauthorised or malicious access (Nath et al, 2018). The edge nodes are the devices which gather data from sensors and interact with physical objects in the ecosystem (Kumar et al, 2017).

SoC design consists of interactive firmware and hardware. In order to verify the security of these modules, a method called instruction-level abstraction (ILA) is used, which addresses verification of properties across firmware and hardware by "Raising the level of abstraction of hardware modules to be similar to that of instructions in firmware" (Malik and Subramanyan, 2016). Hence, this technique helps to ensure that hardware implementations and the firmware of the chip do not contradict with each other. Moreover, there is hardware support for security called Trusted Execution Environment (TEE) for microcontrollers and SoCs used in sensitive applications (Kumar et al, 2017).

Security operations such as encryption-decryption and access blocking are handled at the hardware level (Nichols, 2019). In order to raise security of a chip, cryptography is used. There are two crypto systems, symmetric and asymmetric (Elkeelany and Olabisi, 2006). Symmetric (Schubert and Anheier, 1999) systems use only one key to encrypt and decrypt messages. Asymmetric systems on the other hand, use two that are called public and private keys.They use the public key to encrypt and the private key to decrypt messages (Elkeelany and Olabisi, 2006).

Hitherto, patching has been used only on the software (or firmware) side of the system, in order to strengthen or to cover a discovered security hole. "Hardware patching" means hardware implementation of security requirements that permit seamless post-silicon adaptation (Nath *et al.,* 2018). Recently, it is firmly suggested that also the hardware should be patchable on devices which are to become part of the IoT (Ray et al, 2017). The reason given for this is because soon it will not be possible to fix every single security vulnerability only by altering the software (Ray et al, 2017). In addition to that, implementing a functionality in hardware reduces the energy consumption greatly which is an immense advantage (Ray et al, 2017). Secure hardware systems demand a very high level of verification due to the fact that "patching" is not an option (Bartley, 2017). This can indicate that when patching is made available, enforcements to verify hardware systems may be reduced.

In order to increase patchability of the system, Nath *et al.*2018, introduces a new framework with a centralised Reconfigurable Security Policy Engine (RSPE), smart security wrappers, and Design-for-Debug (DFD) infrastructure interface. Design-for-test (DFT) infrastructure is used to tackle hardware trojan problems with minimum hardware insertion into SoCs (Backer et al, 2014). The DFT infrastructure enables SoC integrators to gain access and have control over IP hardware during post-silicon testing, and after the testing, the DFT infrastructure remains unused (Backer et al, 2014).

The integration of these chips involves connecting third party IP blocks along with DFT hardware pieces, and verification and validation processes of the overall system design (Saleh et al, 2006). These third party hardware IP may be acquired from untrusted vendors and in that case they may have a different level of integrity and security issues (Ray et al, 2018)(Gundabolu and Wang, 2018). These third party vendors may compromise the integrity and security of an SoC design by inserting malicious hardware trojans in their IP (Ray et al, 2018), so the risk of the built-in hardware trojans emerges (Ray et al, 2018)(Li et al, 2016). Unlike software trojans, hardware trojans cannot simply be removed and these trojans may hinder fundamental functionalities of the hardware (Kim and Villasenor, 2015)(Jin and Phatak, 2015). The detection of these trojans occurs by going through post-deployment functional verifications (Jin and Phatak, 2015).

`

Functional Verification Examples

Property Specification Language (PSL) Assertions: They are used to aid verification of complex systems and they may be used to monitor various signals for legal and illegal transitions (Montador, n.d.).

Transaction based testbench-simulations: Traditional behavioural testbenches may become bulky when it comes to testing complex systems (Montador, n.d.) (e.g. going through all the states of a system would be burdensome and cost too much time).

Verification IP: It is used to ensure IP interoperability and system behaviour (Mentor, n.d.). They can also support rapid testbench development (Sondrel, 2016). Usage of IP in a testbench may include functionalities such as keeping track of the traffic between interfaces, and reporting on any violations with respect to existing standards or specifications (Mantador, n.d.).

Trace-Based Debugging: This technique allows SoC integrators to monitor internal signals of IP cores embedded in the SoC (Backer et al, 2016). Its uniqueness is that there is no need to stop the execution in order to capture the values of the signals (Backer et al, 2016).

These are only a few examples of how a hardware can be verified. If we are to talk about standards, the Universal Verification Methodology (ACCELLERA, 2009) is widely used for integrated circuits.

4. Software Security Architecture

Naturally, protecting a chip from hardware faults and producing them without bugs and malfunctions is not solely enough. The SoC has to be protected also from possible threats coming from faulty firmware or the OS kernel and applications (Ray and Jin, 2015).

Simple software designs are commonly used in most security systems because when the system is simple, there is a lower risk of bugs which are a security vulnerability (ARM, 2009). Because multi-threading introduces another layer of software complexity -which is also very hard to test due to timing sensitivity-, many Secure world software implementations may choose to implement single-processor (ARM, 2009). Secure world, also known as TrustZone and Trusted Execution Environment, is a sand-boxed execution environment that has higher privileges than the normal kernel (Vita Development Wiki, n.d.).

Unauthorised access to sensitive assets proposes a critical issue to SoC designs. And modern SoC designs contain numerous assets of this type which must be secure against unauthorised access (Ray and Jin, 2015). The authentication mechanisms that protect these assets are governed by security policies which are also referred to as security architectures in an SoC design (Ray and Jin, 2015).

In order to make justified design choices concerning what and how much to protect, the threats must be well defined in the scope of their risks and the definition of security for that system (ARM, 2009).

In order to protect the system from malicious software, there are some policies introduced as follows:

Access Control: This policy defines which “agent” has access to which asset at different points of the system execution (Ray and Jin, 2015)(Goguen and Meseguer, 1982).

Information Flow: This technique attempts to analyse all the possible interactions and interferences (Goguen and Meseguer, 1982). Naturally, this policy is implemented along with access control policies and even with additional constraints if necessary (Ray and Jin, 2015). (An agent can be a hardware or a software component of an IP)

Liveness: To meet availability requirements, the system has to perform without any pause or “stagnation” throughout its execution which also prevents getting the system caught in deadlocks or livelocks (Basak, 2015). This policy is meant to ensure protection against denial-of-service attacks (Ray and Jin, 2015).

Time of Check vs. Time of Use (TOCTOU): the security objective of an SoC has to be that at any time, the device may only run authenticated firmware(Krstic et al, 2014). In other words, any agent that wants to gain access to a resource has to be authenticated (Basak, 2015) and it is ensured that the authenticated agent is really accessing the resource (Ray and Jin, 2015).

`

Message Immutability: Messages sent between components have to be exactly how they are expected to be (Ray and Jin, 2015).

Redirection and Masquerade Prevention: A message sent from a component to another cannot be delivered to any other component which masquerades itself as the destination component, or the message cannot be duplicated and sent to another component (Ray and Jin, 2015).

Non-Observability: A private message between two components should never be accessible by another IP during its travel (Ray and Jin, 2015).

Furthermore, besides those policies, there are several characteristics introduced within (Ray and Jin, 2015) such as: *Boot Confidentiality* which instructs that no IP may access internal registers during boot, *Firmware Integrity,* that any firmware running on an IP has to be signed or authenticated beforehand which is closely related to the TOCTOU policy.

5. Weaknesses / Vulnerabilities

As mentioned earlier, the IoT is drawing much attention and it is becoming highly prevalently used by numerous companies all around the world. There are already projects such as cloud based smart cities and perhaps in a few decades these projects will be live. Yet even now there are known failures of some IoT systems. Several of them, which are not deliberate violations, are listed below.

— The vulnerability, or more like a leakage, of emergency broadcast systems produced by Acoustic Technology Inc. (ATI) was found by Bastille Security that command packet broadcasts could be captured over the air, even modified and replayed (Sanders, 2018).
— Belkin WeMo, included digital keys in their firmware and the leakage of these keys was a great weakness for hackers to be able to take control of lights and home appliances (Heideman, 2016)(Goodin, 2014).
— Wink sold hubs that could connect to IoT devices. When the installed security certificate expired the whole IoT system malfunctioned/stopped working (Heideman, 2016), or as described in (Barrett, 2015), lobotomised.

These and many more IoT system failures are attributed to the lack of experience and expertise according to Cisco's Australian CTO Kevin Bloch as he mentioned in an interview (Reichert, 2017). It is credibly true for the technical part of failures of IoT systems, when the system is developed by using appropriate designs, standards and policies. But besides the technicality of a project, it should not be forgotten that the organisation itself plays a big role for a project which also affects communications in teams. Of course communication may be considered at an individual level but the culture inside an organisation has a very important impact on it. Thirdly, available tools may have diverse influences such as the familiarity of the team members about particular tools.

6. Novelty

There are numerous publications which discuss many aspects of chips' software and hardware security as referenced throughout the paper. Yet all of these references look at only singular and specific aspect of security. The purpose of this publication is to summarise how securing a chip is approached on both software and hardware during the whole process of making the chip and maintaining it in the course of its lifetime.

References	Year	Title	Hardware/Software-Firmware	Specification
Kumar, S. *et al*	2017	Security Enhancements to System on Chip Devices for IoT Perception Layer	Both	Cursory solutions and no verifications / focus on IoT devices
Malik, S. et al	2016	Specification and Modeling for Systems-on-Chip Security Verification	Hardware	Hardware Security Verification

`

References	Year	Title	Hardware/Software-Firmware	Specification
Ray, S. *et al*	2017	To Secure the Internet of Things, We Must Build It Out of "Patchable" Hardware	Hardware	Hardware Patchability
Nath, A. *et al*	2018	System-on-Chip Security Architecture and CAD Framework for Hardware Patch	Hardware	Hardware Patchability
Kim, L. *et al*	2015	Dynamic Function Verification for System on Chip Security Against Hardware-Based Attacks	Hardware	Hardware Attacks
Bartley, M.	2017	Hardware Security Challenges and Solutions	Hardware	Hardware Protection
Jin, Y. *et al*	2015	Introduction to Hardware Security	Hardware	Hardware Protection
Schubert, A. *et al*	1999	Efficient VLSI Implementation of Modern Symmetric Block Ciphers	Both	Ciphering data
Montador, A.	n.d.	Verification Methodology for Standards-based IP & SOC	Hardware	Hardware Verification
Sondrel	2016	Functional Verification Techniques for your SoC Design	Hardware	Hardware Verification
Rajesvari, M. *et al*	2013	System-on-Chip (SoC) for Telecommand System Design	Hardware	Chip Design
Krstic, S. *et al*	2014	Security of SoC Firmware Load Protocol	Firmware	Firmware Protection
Mentor	n.d.	Mentor Verification IP: Comprehensive verification IP built using advanced methodologies for fastest time to verification sign-off	Both	Verification
Ray, S. *et al*	2015	Security policy enforcement in modern SoC designs	Hardware	Policies
Backer, J. *et al*	2016	Secure and Flexible Trace-Based Debugging of Systems-on-Chip	Hardware/Firmware	Debugging
Ray, S. *et al*	2018	System-on-Chip Platform Security Assurance: Architecture and Validation	Both	Design and Verification
Wang, L. *et al*	2008	System-on-ChipTest Architectures:Nanometer Design For Testability	Hardware/Firmware	Testing Hardware/Firmware

`

References	Year	Title	Hardware/Software-Firmware	Specification
Gundabolu, S. *et al*	2018	On-chip Data Security Against Untrustworthy Software and Hardware IPs in Embedded Systems	Both	Malicious IPs
Elkeelany, O. *et al*	2006	Gaining Extra Crypto-Security using System on Chip Model for RC5	Both	Cryptography
Backer, J. *et al*	2015	Reusing The IEEE 1500 Design for Test Infrastructure For Security Monitoring of Systems-on-Chip	Both	Design and Architecture for Security
Li, H. *et al*	2016	A survey of hardware Trojan threat and defence	Hardware	Hardware Trojans
ARM®	2019	Building a Secure System using TrustZone® Technology	Hardware/Firmware	Chip Design
ACCELLERA	2011	Universal Verification Methodology (UVM) 1.1 User's Guide	Hardware/Firmware	Verification
Basak, A. *et al*	2015	A Flexible Architecture for Systematic Implementation of SoC Security Policies	Hardware/Firmware	Architecture and Policies

7. Conclusion

In conclusion it is very important for an SoC system in general to have security characteristics and to follow appropriate and correct standards and policies. It is always and has always been a good practice to follow globally accepted standards and define which policies to proceed with. The design process of security, and the decisions taken accordingly, highly depend on the imagination of the organisation. Some of the things may be obvious and underestimated or simply missed due to human factors.

It is extremely important to secure microcontrollers and SoCs (whichever is used in the system) because they are the inevitable edge nodes of IoT (Kumar et al, 2017). By all means, protecting these chips mean protecting the whole system.

8. Acknowledgements

We would like to thank Dominic Fennell for his proof reading.

References

ACCELLERA (2011) Universal Verification Methodology (UVM) 1.1 User's Guide.

ARM® (2009) Building a Secure System using TrustZone® Technology. ARM Security Technology.

Backer, J., Hely, D., Karri, R. (2014) Reusing The IEEE 1500 Design for Test Infrastructure For Security Monitoring of Systems-on-Chip. 2014 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT). Amsterdam: IEEE, pp. 52-56.

`

Backer, J., Hely, D., Karri, R. (2016) Secure and Flexible Trace-Based Debugging of Systems-on-Chip. ACM Transactions on Design Automation of Electronic Systems, 22 (2), Article-31.

Barrett, B. (2015) Wink's Outage Shows Us How Frustrating Smart Homes Could Be. [Online] Available at: https://www.wired.com/2015/04/smart-home-headaches/ [Accessed: 03/06/2019]

Bartley, M. (2017) Hardware Security Challenges and Solutions. TVS. [Online] Available at: https://www.testandverification.com/wp-content/uploads/TVS-Hardware-Security-Challenges-and-Solutions.pdf [Accessed: 21/05/2019]

Basak, A., Bhunia, S., Ray, S. (2015) A Flexible Architecture for Systematic Implementation of SoC Security Policies. 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). Austin, TX, USA: IEEE, pp. 536-543.

Davis, M. (2012) What is an SoC? Hint: the "S" stands for Server. ARM SERVERS [Online] Arm Ltd. Available from: https://armservers.com/2012/05/14/what-is-an-soc-hint-the-s-stands-for-server/ [Accessed: 29/03/2019]

Elkeelany, O., Olabisi, A. (2006) Gaining Extra Crypto-Security using System on Chip Model for RC5. Proceedings of the 38th Southeastern Symposium on System Theory. Tennessee Technological University, TN, USA.

Evans, D. (2011) The internet of things - how the next evolution of the internet is changing everything. White Paper. Cisco Internet Business Solutions Group (IBSG), 2011.

Gelsinger, P. (2006) Moore's Law - The Genius Lives On. *IEEE solid-state circuits society newsletter* September 2006. Available from:http://web.archive.org/web/20070713083830/http://www.ieee.org/portal/site/sscs/menuitem.f07ee9e3b2a01d06bb9305765bac26c8/index.jsp?&pName=sscs_level1_article&TheCat=2165&path=sscs/06Sept&file=Gelsinger.xml [Accessed: 29/03/2019]

Goguen, J., Meseguer, J. (1982) Security Policies and Security Models. IEEE Symposium on Security and Privacy, pp. 11–20.

Goodin, D. (2014) Password leak in WeMo devices makes home appliances susceptible to hijacks (updated). [Online] Available at: https://arstechnica.com/information-technology/2014/02/password-leak-in-wemo-devices-makes-home-appliances-susceptible-to-hijacks/ [Accessed: 03/06/2019]

GSM Association (2014) Understanding the Internet of Things (IoT). [Online] Available from: https://www.gsma.com/iot/wp-content/uploads/2014/08/cl_iot_wp_07_14.pdf [Accessed: 12/04/2019]

Gundabolu, S., Wang, X. (2018) On-chip Data Security Against Untrustworthy Software and Hardware IPs in Embedded Systems. 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 644-649.

Heideman, P. (2016) Early IoT Missteps - 10 Examples of Failure. [Online] Available at: https://www.omniresources.com/blog/early-iot-missteps-10-examples-of-failure [Accessed: 03/06/2019]

Jin, Y., Phatak, D.(ed) (2015) Introduction to Hardware Security. Electronics. Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, USA.

Li, H., Liu, Q., Zhang, J. (2016) A survey of hardware Trojan threat and defence. INTEGRATION, the VLSI journal, 55 (2016) , pp. 426–437.

Kim, L., Villasenor, J. (2015) Dynamic Function Verification for System on Chip Security Against Hardware-Based Attacks. IEEE Transactions on Reliability, 64 (4), pp. 1229-1242.

Krstic, S., Yang, J., Palmer, D., Osborne, R., Talmor, E. (2014) Security of SoC Firmware Load Protocol. IEEE HOST.

`

Kumar, S., Sahoo, S., Mahapatra, A., Swain, A., Mahapatra, K. (2017) Security Enhancements to System on Chip Devices for IoT Perception Layer. 2017 IEEE International Symposium on Nanoelectronic and Information Systems. Rourkela.

Maglaras, L. A., Kim, K. H., Janicke, H., Ferrag, M. A., Rallis, S., Fragkou, P., ... & Cruz, T. J. (2018). Cyber security of critical infrastructures. ICT Express, 4(1), 42-45.

Malik, S., Subramanyan, P. (2016) Specification and Modeling for Systems-on-Chip Security Verification. Department of Electrical Engineering, Princeton University, Texas, USA.

Mentor (n.d.) Mentor Verification IP: Comprehensive verification IP built using advanced methodologies for fastest time to verification sign-off. A Siemens Business. [Online] Available at: https://www.mentor.com/products/fv/verification-ip [Accessed: 22/05/2019]

Montador, A. (n.d.) Verification Methodology for Standards-based IP & SOC. Cadence Design Systems Inc, Livingston, Scotland. [Online] Available at: https://www.design-reuse.com/articles/13229/verification-methodology-for-standards-based-ip-soc.html [Accessed: 22/05/2019]

Nath, A., Ray, S., Basak, A., Bhunia, S. (2018) System-on-Chip Security Architecture and CAD Framework for Hardware Patch. IEEE. USA.

National Cable & Telecommunications Association - NCTA (2015) Behind The Numbers: Growth in the Internet of Things [Online] Available at: https://www.ncta.com/whats-new/behind-the-numbers-growth-in-the-internet-of-things [Accessed: 12/05/2019]

Nichols, M. (2019) Is System-on-Chip a viable solution for IoT security? [Online] Born2Invest. Available from: https://born2invest.com/articles/system-chip-viable-solution-iot-security/ [Accessed: 31/05/2019]

Nolan, S. (n.d.) Power Management for Internet of Things (IoT) System on a Chip (SoC) Development. Vidatronic, Inc. [Online] Available at: https://www.design-reuse.com/articles/42705/power-management-for-iot-soc-development.html [Accessed: 18/05/2019]

Palomar Technologies (n.d.) System on a Chip (SoC) [Online] Available at: https://www.palomartechnologies.com/applications/system-on-a-chip [Accessed: 11/05/2019]

Rajesvari, R., Manoj, G., Angelin Ponrani, M. (2013) System-on-Chip (SoC) for Telecommand System Design. International Journal of Advanced Research in Computer and Communication Engineering, 2 (3), pp. 1580-1585

Ray, S., Jin, Y. (2015) Security policy enforcement in modern SoC designs. 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). Austin, TX, USA: IEEE.

Ray, S., Basak, A., Bhunia, S. (2017) To Secure the Internet of Things, We Must Build It Out of "Patchable" Hardware. IEEE Spectrum. [Online] Available at:https://spectrum.ieee.org/telecom/security/to-secure-the-internet-of-things-we-must-build-it-out-of-patchable-hardware [Accessed: 12/05/2019]

Ray, S., Peeters, E., Tehranipoor, M., Bhunia, S. (2018) System-on-Chip Platform Security Assurance: Architecture and Validation. *Proceedings of The IEEE,* 106 (1), pp. 21-37.

Reichert, C. (2017) Cisco: Most IoT projects are failing due to lack of experience and security. [Online] Available at: https://www.zdnet.com/article/cisco-most-iot-projects-are-failing-due-to-lack-of-experience-and-security/ [Accessed: 03/06/2019]

Saleh, R., Wilton, S., Mirabbasi, S., HU, A., Greenstreet, M., Lemieux, G., Pande, P., Grecu, C., Ivanov, A. (2006) System-on-Chip: Reuse and Integration. Proceedings of the IEEE, 94 (6), pp. 1050-1069

Sanders, J. (2018) 5 Biggest IoT Failures in 2018. [Online] Available at: https://www.techrepublic.com/article/5-biggest-iot-security-failures-of-2018/ [Accessed: 03/06/2019]

`

Schubert, A., Anheier, W. (1999) Efficient VLSI Implementation of Modern Symmetric Block Ciphers. Proceedings of ICECS'99, Pafos, Cyprus 1999.

Sondrel (2016) Functional Verification Techniques for your SoC Design. [Online] Available at: http://blog.sondrel.com/functional-verification [Accessed : 22/05/2019]

Vita Development Wiki (2016) Secure World. [Online] Available at: https://wiki.henkaku.xyz/vita/Secure_World [Accessed: 02/06/2019]

Wang, L., Stroud, C., Touba, N. (2008) System-on-Chip Test Architectures:Nanometer Design For Testability. Burlington, USA: Morgan Kaufmann Publishers.

`